**Introduction**

A generally acknowledged "best practice" for simplifying the management of network infrastructures is to reduce the number of different types of devices and software versions that are deployed. However, no single family of devices can scale to meet the diverse needs of the wiring closets, data centers and network cores across the range of sites typically found in large organizations. As a result, the generally recommended approach to dealing with multiple switch platforms is based on the following principles:

- Consolidate networks by reducing the number of equipment vendors and the number of different models of devices from each vendor
- For each switch or router vendor, minimize the number of OSs and OS releases in use

Even when these principles are diligently applied, network managers still must frequently deal with multiple, divergent network operating systems, even from the same equipment vendor. Multiple network operating systems deployed across the network adds considerable complexity to a wide range of management tasks, including:

- Patch management
- Tracking new releases, features/feature changes and security alerts
- Managing ACLs and roles-based access control (RBAC)
- Interfacing with diverse CLIs, MIBs, traps, diagnostics and instrumentation features

Force10 Networks has addressed the problem of network OS complexity by leveraging the concept of hardware abstraction to port its modular Force10 Operating System (FTOS) from the E-Series switch/router platform to both the C-Series and S-Series platforms. For all Force10 Networks families of switch/routers, the FTOS software is compiled from a single common base of source code. This development gives network managers the option of running a single operating system across all the switches and routers in the end-to-end Layer 2/Layer 3 network infrastructure from the access/aggregation tier to the network core and the data center.

With FTOS deployed end-to-end across the network, network managers have the ultimate in flexibility to simplify operation and management of the network. Reducing the complexity of the network operating system environment has a number of advantages:

- Improved network reliability and predictability
- Fewer discontinuities in features/functions across network tiers
- Better interoperability among devices
- Simplified patch processes and version management
- Less expertise required in navigating disparate user interfaces (e.g., CLIs)
- Simplified troubleshooting and fault management
- Simpler recovery from security vulnerabilities and intrusions
The portability of FTOS across multiple platforms – the E-Series, C-Series and S-Series – is made possible by Force10's implementation of the hardware abstraction layer (HAL) code in FTOS. With HAL, porting FTOS to a new hardware platform can be accomplished without rewriting the core of FTOS (i.e., the modular processes and kernel) that comprises the majority of the code base. New code that is required for porting is decoupled from the core code by the layer of abstraction and is restricted to hardware-specific interfaces to HAL.

A software release is common to all three platforms, with each platform running an image specific to the hardware technology used in the system. When a new software version is released, all new software features are compiled and automatically integrated into each platform at the same time, without the need for multiple branches and merges in the source code. Any new hardware-specific feature is enabled by HAL and are incorporated at compile time if the underlying hardware supports it.

The remainder of this white paper describes the role that HAL plays in FTOS and provides a discussion of the benefits that result from OS portability across the entire Force10 switch/router product line, resulting in a single switch/router OS that can span the entire network.

What is a Hardware Abstraction Layer?

The hardware abstraction layer (HAL) is a layer of software that decouples the kernel of an operating system from the specific details of the underlying hardware. By hiding the differences in underlying hardware, the HAL isolates the operating system kernel source code from a major rewrite whenever there are changes in the hardware platform.

In the realm of computers, Mac OS X, BSD Unix, Windows, Solaris and other portable operating systems employ the concept of hardware abstraction. Hardware abstraction allows these OSs to be readily adapted to employ a wide range of hardware subsystems (e.g., storage, sound or video), run on different generations of the same microprocessor architecture, and even be ported to completely different microprocessor architectures, as was recently the case for Mac OS X.

NetBSD and HAL

Force10's FTOS is based on NetBSD, a highly portable, open source, modular operating system that has been optimized for networking applications. NetBSD is widely noted for its clean design and advanced features, including the hardware abstraction that allows it to be highly portable and extensible to new hardware variants. NetBSD's hardware abstraction makes a clean distinction between chipset drivers and the drivers for peripheral devices, allowing the same device-driver source code to be used across different CPU architectures and bus types. NetBSD has been ported to over 50 different hardware platforms including large-scale server systems, desktop systems, handheld devices, and the control plane CPUs of numerous network elements, such as switch/routers and firewalls. The OS is ported to new hardware by creating the required driver interfaces to the hardware.

FTOS and HAL

Figure 1 shows Force10's adaptation of the NetBSD architecture for FTOS, the operating system shared...
by the E-Series, C-Series and S-Series family of switch/routers. In FTOS, separate modular processes have been created for:

- Each of the Layer 2 protocols (STP, LACP, etc.)
- Each of the Layer 3 routing protocols (IS-IS, OSPF, BGP, static routing)
- Various system services and management functions (SNMP, CLI, etc.)
- Security services and protocols (SCP, SSH, TACACS+/RADIUS, ACLs)

Whenever these processes need to share information, the exchanges are all channeled through a robust inter-process communication (IPC) mechanism (not shown in this diagram) layered on top of the kernel.

The FTOS HAL implementation provides the abstraction of the capabilities of the different hardware platforms. Figure 1 shows a representation of the FTOS source code that provides cross platform support. When the FTOS source is compiled for a particular platform, the OS process modules and drivers are trimmed to meet the specific requirements of the target platform. Compile time optimization of FTOS/HAL improves performance and minimizes system resource requirements. This allows FTOS to easily scale from a multi-slot modular switch/router down to a 1 RU fixed configuration switch/router.

The hardware abstraction software layer within FTOS has numerous advantages that contribute significantly to the efficiency and effectiveness of Force10’s product development and support teams. The following benefits are reflected in the competitiveness of Force10’s switch/routers and also lead to many of the end user benefits described in the next section of this document.

- **Product/Feature Roll-Out:** With only one track of switch/router OS development, there is a linear, sequential code path, as shown in Figure 2, rather than a code tree with multiple branches and parallel paths of code and functionality evolution. A single code track translates to a single, accelerated roadmap for the roll out of new features. In addition, programmer productivity is improved by cross platform development and only a single control plane architecture team is required for all families of switch/routers. Productivity improvements also facilitate scaling the product line by adding support for new hardware platforms.

- **Release Quality and Testing:** All testing resources can be applied to a single OS release train, greatly enhancing the efficiency and speed of the entire test cycle including regression and performance testing.

- **Guaranteed Protocol Interoperability across Platforms:** With only one Layer 2/Layer 3 protocol stack across all switch/router models and platforms, there is no requirement for extensive testing to verify interoperability.

- **Security Vulnerabilities:** Security issues are easier to resolve within a single code base rather than in a plethora of parallel OS releases.

- **Product Support:** The simplicity of a single code base streamlines diagnostics, troubleshooting, product TAC training, and software management and upgrades because only one patch process and one bug-fix process are required to support multiple product families.

**User Benefits of HAL-enabled FTOS**

The HAL-enabled commonality of source code across all FTOS switch/routers leads to numerous end user benefits, including:

- **Higher Reliability/Availability:** The common base of code results in highly reliable, more thoroughly tested software. Higher levels of availability are achieved because of fewer software errors and system restarts. In addition, system downtime due to operator errors is reduced because of the commonality of configuration management functionality and user interfaces across the switch/router product spectrum.

- **Consistent Features and Functionality:** A consistent set of switch/router features throughout the network
infrastructure from the data center and core to the wiring closet and branch office simplifies network design and facilitates consistent implementation of end-to-end QoS and security policies. Table 1 shows the range of consistent, cross-platform FTOS functionality as well as some of the features functions that are platform-specific because of constraints in the underlying hardware.

- **Feature Ubiquity/Immediacy:** Newly developed common features become available on all platforms on the same day. Therefore, planning, staging and deploying OS upgrades to take advantage of new features are greatly simplified.

- **Protocol Interoperability:** A single common L2/L3 protocol stack assures interoperability across all platforms.

- **Simplified Management/Administration:** A common CLI and a common XML management model across all models of switch/router simplifies network management and provides the opportunity to automate management functions. In addition, life cycle management of the switched infrastructure is streamlined by greatly simplified patching and upgrade processes, which do not need to deal with the complexity of multiple OS versions evolving on parallel paths.

- **Common RBAC and Security Model:** All systems can now utilize the exact same access control, ACL, and roles-based security frameworks across all platforms allowing the deployment of a single, comprehensive, network-wide security model.

- **Common Serviceability:** A common serviceability model allows consistent troubleshooting procedures and hardware diagnostics to be performed across all platforms.

- **Lower OpEx and TCO:** A switch/router software environment that features higher reliability and more consistent functionality and management capabilities contributes to a significant reduction in operational expense, which is a major component of the total cost of ownership (TCO) for the network.

---

**Design Challenges Posed by Hardware Abstraction**

The primary challenge in designing a hardware abstraction layer is to achieve the proper partitioning of code between the cross-platform code that is positioned above the HAL, as in Figure 1, and the platform-specific code that is positioned in the drivers that are the NetBSD kernel. The partitioning must maximize the amount of cross-platform code, while still providing access to the full capabilities of each hardware platform.

Another design challenge encountered when a single operating system is run over a highly scalable set of platforms is ensuring that the common operating system is compatible with the CPU and memory resources available in compact (e.g., 1 RU) switch platforms. FTOS deals with this challenge by using a compiled version of the common source code for each hardware platform. With this approach, each platform needs to support only the applicable cross-platform and platform-specific code modules. Compilation is further leveraged in modular platforms (such as the E-Series) that include multiple processors distributed across the RPM and the line cards. The multi-processor architecture allows the creation of a server/agent relationship between the RPM processors and the line card processors. Each RPM processor’s FTOS compilation needs to include only the required cross-platform functional modules, while the line card processor’s FTOS version includes only the appropriate platform-specific modules plus the cross-platform modules needed to support the server/agent relationship.

A third challenge is presented each time the operating system is modified to support a new hardware platform. As more hardware platforms are added, the inevitable differences in functionality tend to shrink the cross-platform code while expanding the platform-specific code. Adding new platforms with unique functionality also creates challenges in maintaining a common user interface (CLI) in the face of the need for new commands to access the new functionality.

To address this challenge, Force10 engineering has implemented a methodology that starts from consistency in every instance. Where possible, no deviations to the common CLI method should be allowed, and, as needed, deviations should be specific to the platform capabilities and bounded solely by those hardware or scale differences.
Conclusion

Today’s network administrator is facing a number of challenges, such as:

- Increased focus on reliability and manageability because of greater business reliance on the network
- Assimilation of real-time applications (VoIP and video)
- Assuring satisfactory application performance with QoS
- Scaling the network to meet escalating bandwidth requirements
- Ensuring security in the face of a more fluid network environment

All of these challenges are considerably easier to meet where the underlying network infrastructure has been consolidated to minimize overall network complexity, including the complexity implicit with a diversity of network operating systems.

With HAL-enabled FTOS running across the entire spectrum of Force10 Networks’ switch/routers (E-Series, C-Series and S-Series), it is now possible to maximize the consolidation of the end-to-end switched infrastructure in all three critical dimensions:

1. Network hardware for supporting simplified network topologies with unmatched scalability for 10 GbE, and eventually 40 and 100 GbE.
2. Network OS software with FTOS scaling across the wiring closet, data center and core of the network.
3. Network management with a single interface and management framework spanning the entire network.

<table>
<thead>
<tr>
<th>Feature</th>
<th>E-Series</th>
<th>C-Series</th>
<th>S-Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unix-like NetBSD modular OS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>One source tree for all platforms</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>One FTOS version for all platforms</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Modular feature development and QA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature parity between FTOS versions</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Industry standard CLI with ease of use features</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Identical CLI on all platforms</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>High-end management and CLI features</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>In-service debugging and diagnostic features</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1. Cross platform features/functionality of FTOS enabled by HAL